УДК 681.3.06

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИЕРАРХИЧЕСКИХ АРХИТЕКТУР ПАРАЛЛЕЛЬНЫХ СИСТЕМ БАЗ ДАННЫХ*

П.С. Костенецкий, Л.Б. Соколинский
Челябинский государственный университет, Россия

В работе предлагается новый подход к моделированию иерархических аппаратных архитектур параллельных систем баз данных и описывается программный комплекс, основанный на этом подходе. Данный программный комплекс позволяет исследовать эффективность различных иерархических конфигураций, включая гибридные, в контексте задач баз данных класса OLTP. Это дает возможность исследовать широкий спектр перспективных гибридных и иерархических кластерных архитектур без больших затрат на их аппаратную реализацию.

Введение

В последнее десятилетие возник целый ряд задач, требующих хранения и обработки сверх больших объемов данных [3]. В связи с этим, является актуальной задача разработки новых иерархических архитектур многопроцессорных систем баз даны, позволяющих обрабатывать большие объемы информации за малое время. Наибольший интерес здесь вызывают гибридные иерархические архитектуры [1]. Исследование подобных архитектур затруднено, так как практическое конструирование мультипроцессоров требует больших финансовых затрат, связанных с приобретением и реконфигурацией дорогостоящего оборудования. Поэтому для большинства исследовательских организаций данный подход оказывается, как правило, неприемлемым. В соответствии с этим, перспективной является задача разработки моделей представления многопроцессорных систем баз данных, которые позволяли бы исследовать различные многопроцессорные конфигурации без их аппаратной реализации. Исследование многопроцессорных систем баз данных с одноуровневой и двухуровневой архитектурой производилось при помощи моделей во многих работах, (см., например, [5]), однако, в общем виде гибридные иерархические архитектуры не исследовались.

В данной работе предлагается подход к моделированию иерархических архитектур параллельных систем баз данных с произвольным количеством уровней и описывается программный комплекс, основанный на этом подходе, позволяющий исследовать различные иерархические конфигурации, включая гибридные. Это дает возможность исследовать широкий спектр перспективных гибридных и иерархических кластерных архитектур без больших затрат на их аппаратную реализацию.

1. Модель многопроцессорной системы баз данных

Нами предлагается *DM(Database Multiprocessor) - модель*, которая может применяться для исследования широкого спектра аппаратных архитектур, в том числе гибридных. *DM*-модель разработана для систем параллельной обработки транзакций в режиме OLTP и опирается на реляционную модель данных [6].

В рамках предлагаемой модели, моделируемые архитектуры представляются в виде DM-дерева.

DM-дерево - это граф, вершины которого относятся к одному из трех классов: процессорные модули, дисковые модули и модули сетевых концентраторов. Далее мы

* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 03-07-90031).

будем называть узлы *DM*-дерева модулями. Ребра графа, в данном контексте, являются абстракцией двунаправленных каналов передачи данных, соединяющих модули. Корнем *DM*-дерева может быть только модуль сетевого концентратора. Процессорный модуль может быть соединен с дисковым модулем только через сетевой концентратор. Дисковые и процессорные модули всегда являются листьями *DM*-дерева.

Будем называть представление реальной мультипроцессорной системы в виде *DM*-дерева *виртуальным мультипроцессором*.

В рамках нашей модели обработка данных выполняется с точностью до гранулы, называемой *пакетом*. В качестве пакета может фигурировать одна или несколько строк реляционной таблицы.

В рамках *DM*-модели введем понятие такта следующим образом: *такт* – это промежуток времени, необходимый для того, чтобы все процессорные модули обработали один пакет, дисковые модули произвели чтение-запись данных одного пакета, а модули сетевых концентраторов обработали все пакеты, ожидающие передачи в данный момент времени.

Время обработки запроса в *DM*-модели вычисляется как суммарная длительность тактов выполненных системой в промежутке времени между инициализацией и завершением запроса. Данное огрубление понятия времени обработки запроса является приемлемым для OLDP-приложений, так как мы можем считать, что размер пакета несоизмеримо мал по сравнению с размером базы данных.

С каждым модулем *DM*-дерева мы связываем некоторый *коэффициент трудоемкости* h_{n_i}, который определяет время, необходимое данному модулю для обработки одного пакета данных.

Для OLTP-приложений время обработки процессорным модулем одного пакета в 10^5-10^6 раза быстрее, чем время чтения-записи пакета дисковым модулем или его передача через сетевой концентратор [7]. В соответствии с этим, мы полагаем в *DM*-модели, что коэффициент трудоемкости для процессорного модуля всегда равен нулю.

Так как модуль сетевого концентратора n_i за один такт может передавать одновременно несколько пакетов, то для каждого модуля сетевого концентратора, кроме коэффициента трудоемкости, мы вводим функцию помех $f_{n_i}(m)$, в которой m - число одновременно проходящих через модуль пакетов. Таким образом, время, требуемое модулю сетевого концентратора для выполнения одного такта, вычисляется по следующей формуле:

$$t_{n_i} = h_{n_i} + f_{n_i}(m).$$

Обозначим N - множество всех модулей сетевых концентраторов *DM*-дерева, *M* - множество всех дисковых модулей *DM*-дерева, *S=(N,P,D)* - множество всех модулей *DM*-дерева. Посредством n_i, p_i, d_i мы будем обозначать элементы соответствующих множеств, то есть, модули сетевых концентраторов, процессорные модули и дисковые модули.

Время, затрачиваемое системой на выполнение одного такта, вычисляется по формуле:

$$t_i = \max(\max(t_n), \max(t_d)), \forall n \in N, \forall d \in D.$$

В соответствии с этим, общее время работы системы, затраченное на обработку смеси транзакций объемом в k тактов, вычисляется по формуле:

$$T = \sum_{i=1}^{k} t_i$$

В рамках предлагаемой модели такт делится на 2 этапа:

1) процессоры производят по одному обращению к дискам;
2) модули сетевых концентраторов передают пакеты на смежный уровень.

Каждый такт все процессорные модули могут производить обращения к дисковым модулям. Существует два типа обращений: чтение с диска и запись на диск. Различия между данными типами обращений заключается в следующем: при чтении с диска, пакет информации появляется и начинает свое движение по дереву архитектуры со стороны дискового модуля по направлению к процессору, который произвел обращение. В случае записи на диск, в обратном направлении. Путь доставки пакета вычисляется динамически от текущего местоположения, по детерминированному алгоритму [2].

При данной реализации, основной характеристикой моделируемых архитектур является время, за которое выполнилось заданное число тактов.

2. Методы моделирования

Данный раздел посвящен методам моделирования, используемым при реализации ЭВМБД (Эмулятора Виртуальных Мультипроцессоров Баз Данных).

Для реализации программной системы «ЭВМБД» был выбран язык С.

Эмулятор виртуальных мультипроцессоров баз данных строится из следующих виртуальных устройств: *процессорные модули, дисковые модули, модули сетевых концентраторов.* В подразделах 2.1.1–2.1.3 рассмотрены методы моделирования объектов указанных выше типов.

2.1.1 Процессорные модули

Для каждого процессорного модуля задан список процессов. Процессы в данном контексте имитируют обработку процессорным модулем запросов к базе данных.

Для каждого процесса установлена своя вероятность срабатывания. Для эмуляции простоя процессора, в списке процессов находится специальный процесс: «простой процессора», для которого так же установлена вероятность срабатывания. За каждый такт работы системы, в каждом процессорном модуле срабатывает один процесс из списка, исключая окончившие работу процессы. Процесс производит единственное обращение к одному из дисков системы [2].

2.1.2 Дисковые модули

Дисковый модуль каждый такт выполняет следующий простой алгоритм:

1) Если во входном буфере есть пакеты, то удалить один пакет из входного буфера и из списка ожидаемых пакетов. Таким образом, эмулируется обработка диском пришедшего пакета.
2) Если на диске имеются не отправленные пакеты, то все они отправляются во входной буфер вышестоящего сетевого концентратора.

2.1.3 Модули сетевых концентраторов

Модули сетевых концентраторов пересылают пакеты информации, создаваемые при обращениях, от отправителей к адресатам. В рамках нашей модели, сетевые концентраторы могут производить отправку пакетов только на непосредственно примыкающие модули. Путь доставки пакетов от отправителя к адресату заранее неизвестен. Поэтому, каждый такт модули сетевых концентраторов вычисляют порт, на который следует отправить каждый из пришедших пакетов. За такт все пакеты должны быть обработаны один раз, т.е. за такт пакет может переместиться только на один уровень вверх или вниз по *DM*-дереву. После обработки модулем сетевого концентратора, пакет помечается как обработанный на данном такте.

Таким образом, путь доставки пакетов от отправителя к адресату вычисляется динамически, исходя из текущего местоположения пакета. Вычисление пути происходит по детерминированному алгоритму [2]. Абстрактно этот алгоритм можно описать так:

1) подняться до корня минимального поддерева,
2) спуститься до адресата.

Если через сетевой концентратор за один такт должны одновременно пройти несколько пакетов, то к времени, требуемому модулю сетевого концентратора для выполнения одного такта, прибавляется функция помех:

$$f(m) = e^{\frac{m}{\delta}}$$

В данной формуле: m – число одновременно проходящих через концентратор пакетов, δ – параметр.

Заключение

В работе был описан подход к моделированию иерархических архитектур параллельных систем баз данных. На базе описанного подхода был разработан программный комплекс, получивший название ЭВМБД (Эмулятор Виртуальных Мультипроцессоров Баз Данных). ЭВМБД позволяет моделировать практически любые архитектуры параллельных систем баз данных. Это дает возможность исследовать широкий спектр перспективных архитектур, в том числе гибридных иерархических кластерных архитектур, без больших затрат на их аппаратную реализацию. С помощью разработанного программного комплекса можно выполнять моделирование работы многопроцессорных систем баз данных на вычислительной системе с любым количеством процессоров, в том числе и на однопроцессорной. Данный подход был применен для моделирования иерархических гибридных архитектур класса CDN [1]. Предварительные эксперименты подтвердили адекватность и эффективность предложенной модели.

Литература

1. Соколинский Л.Б. Классификация и анализ параллельных архитектур систем баз данных // Алгоритмы и программные средства параллельных вычислений: [Сб. науч. Тр.]. -Екатеринбург: УрО РАН. -2003. -Вып. 7. -С. 185-216.

2. Костенецкий П.С., Соколинский Л.Б. Моделирование и анализ иерархических архитектур параллельных систем баз данных. Технический отчет OMEGA011. ЧелГУ, 2004 (http://www.csu.ru/~sok/papers/omega_rep/11.html).

3. Соколинский Л.Б. Параллельные машины баз данных // Природа. Естественно-научный журнал Российской академии наук. 2001. No. 8. С. 10-17.

4. Соколинский Л.Б. Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической архитектурой // Программирование. -2001. -No. 6. -С. 13-29.

5. Bhide A. An Analysis of Three Transaction Processing Architectures // Fourteenth International Conference on Very Large Data Bases (VLDB'88), August 29 - September 1, 1988, Los Angeles, California, USA, Proceedings. Morgan Kaufmann. 1988. P. 339-350.

6. Кодд Е.Ф. Реляционная модель для больших совместно используемых банков данных // СУБД. 1995. N1. С. 145-169.

7. Gray J., Graefe G. The Five-Minute Rule Ten Years Later, and Other Computer Storage Rules of Thumb // SIGMOD Record. -1997. -Vol. 26, No. 4. -P. 63-68.

8. Байкова И., Кулагин М. Современные дисковые системы RAID // Открытые системы. 1995. №3. С. 50-55.